

INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO



Aplicación para comunicaciones en red  
3CM17

Practica 05

"WGET"

Integrantes:

- ♥ Bocanegra Heziquio Yestlanezi
- ♥ Martinez Cruz José Antonio


Profesor: Moreno Cervantes Axel Ernesto





## índice

Objetivo.....	3
Introducción.....	3
WGet.....	3
Hilo .....	4
Peticiónes HTTP .....	5
Clasificación de las peticiónes HTTP.....	5
Peticiónes HTTP Safe .....	5
Peticiónes HTTP Idempotent .....	5
Desarrollo .....	6
Conclusiones .....	8
Bocanegra Heziquio Yestlanezi.....	8
Martínez Cruz José Antonio .....	8





## Objetivo


El estudiante implementará una aplicación para la descarga de dominios mediante una alberca de hilos y el protocolo HTTP.

## Introduccion

El envío de archivos a través de la red es una característica importante para la gran mayoría de las aplicaciones que hoy día se utilizan (blogs, redes sociales, mensajería instantánea, Declaración de impuestos, educación en línea, etc.), sin embargo, no todas las aplicaciones disponibles permiten el envío de archivos de gran tamaño (p.e. El correo electrónico no permite enviar archivos de más de 10 o 20 MB). Esto hace necesario el desarrollo de aplicaciones que permitan transferir archivos sin importar el tamaño de éstos.

## WGet

GNU Wget es una herramienta libre que permite la descarga de contenidos desde servidores web de una forma simple. Su nombre deriva de World Wide Web (w), y de «obtener» (en inglés get), esto quiere decir: obtener desde la WWW. Fue escrito originalmente por Hrvoje Nikšić y por ser un proyecto de software libre tiene una gran cantidad de colaboradores directos e indirectos. Actualmente admite descargas mediante los protocolos HTTP, HTTPS y FTP. Entre las características más destacadas que ofrece Wget está la posibilidad de fácil descarga de mirrors (repositorios) complejos de forma recursiva, conversión de enlaces para la visualización de contenidos HTML localmente, soporte para proxies, etc. Su primera versión se lanzó en 1996, coincidiendo con el boom de popularidad de la web. Es un programa utilizado a través de línea de comandos, principalmente en sistemas tipo UNIX, especialmente en GNU/Linux. Escrito en el lenguaje de programación C, Wget puede ser fácilmente instalado en sistemas derivados de UNIX, y ha sido portado a muchas interfaces gráficas de usuario (GUI) y aplicaciones gráficas de descarga como Gwget2 para GNOME, wGetGUI3 y VisualWget4 para Microsoft Windows, Wget 1.10.2r25 para Mac OS X.





## Hilo


Un hilo (en inglés "thread") es la menor de las estructuras lógicas de programación que se ejecuta de forma secuencial por parte del planificador del sistema operativo (nota: forma secuencial no quiere decir que no haya bucles, solo que es una línea secuencial de ejecución: una sentencia después de otra).

Los hilos son más "ligeros" que sus hermanos mayores (los procesos) ya que muchos de los recursos que es necesario reservar para lanzar y ejecutar un proceso, son compartidos entre distintos hilos dentro de un mismo proceso.

Los hilos existían como recurso de programación desde mucho tiempo antes de que empezaran a hacerse comunes los procesadores de varios núcleos y otras estructuras de paralelización habituales en las arquitecturas de hoy día (pipelines, grids, GPUs, etc.). Pero no nacieron como forma de acelerar la ejecución de los programas, sino como forma de "simplificar" la forma en la que se diseñaban los programas más complejos. Bueno, se diseñaban... y se siguen diseñando.

Así, cuando un desarrollador necesita que un programa ejecute diversas funciones dando la sensación de que todas ellas se ejecutan casi de forma instantánea, lo que hace es dividir la ejecución en distintos hilos relativamente simples, cada uno con una función claramente definida. Luego se deja que sea el planificador del sistema operativo (o de la máquina virtual) el que vaya ejecutando un trocito de código de cada hilo, y conmute entre uno y otro hilo dentro del mismo proceso para obtener la sensación de que las diversas funciones del programa se ejecutan "a la vez".

El ejemplo que siempre se pone es el de un procesador de textos que se diseña con diversos hilos: uno se encarga de recoger los caracteres de teclado, otro de dibujar la información en pantalla, otro de hacer el análisis de la ortografía del documento y uno más de volcar cada cierto rato a disco una copia de seguridad del documento. Todos los hilos, se ejecutan "a la vez" usando los recursos del mismo proceso (y en realidad usando el mismo procesador) creando la ilusión de que el proceso realiza varias tareas a la vez. El programa (que en la jerarquía del sistema operativo corresponde con un proceso) se divide en varios hilos y los hilos se suelen comunicar entre sí mediante mecanismos relativamente sencillos (normalmente zonas de memoria compartida protegidas por semáforos o monitores).





## Peticiones HTTP

HTTP contiene un grupo de peticiones HTTP (también llamadas HTTP verbs por, el tipo de nombre que manejan casi todos ellos -pues si bien algunos son sustantivos, la gran mayoría no-) que nos ayudan a especificar la acción que se requiere realizar en un elemento determinado y aunque estas peticiones tienen distintas semánticas, también tienen muchas similitudes en las mismas que evitan que este grupo se extienda demasiado.

## Clasificación de las peticiones HTTP

Estas peticiones las podemos clasificar en Safe e Idempotent:


### Peticiones HTTP Safe

Un método HTTP es considerado safe o seguro si no altera el estado del servidor. En otras palabras, un método es seguro si conduce a una operación de 'sólo lectura'. Algunos de los métodos HTTP más comunes son seguros: OPTIONS, GET o HEAD. Todos los métodos seguros son también a su vez idempotent (así como también algunos, pero no todos, los métodos inseguros como DELETE o PUT).

### Peticiones HTTP Idempotent

Así como un objeto cualquiera tiene la propiedad de idempotencia si al realizar una operación muchas veces da el mismo resultado cual si se hubiese realizado la operación una sola vez, un método HTTP es idempotente si una solicitud idéntica puede realizarse una o demasiadas veces consecutivamente obteniendo el mismo resultado dejando al servidor en el mismo estado.

En la vida y en los números reales, podríamos llamar al 1 y al 0 como los únicos idempotentes para la operación de multiplicación, pues estos números al multiplicarse en muchas ocasiones, da como resultado él mismo ( $1^n = 1$ ). Volviendo a HTTP, los métodos que (implementados correctamente) son idempotentes son el GET, HEAD, PUT y DELETE y como lo explicamos en los métodos Safe, todos éstos últimos son idempotentes.





## Desarrollo


- El usuario especificará una URL de un sitio web; a continuación, la aplicación tratará de visitar dicha dirección web y copiar su contenido en un archivo local.
- A continuación, se validará si la página tiene hipervínculos y los descargará de forma recursiva (también modificará los hipervínculos para que puedan ser accedidos localmente). Para realizar esta tarea de descarga se usará un conjunto de hilos (thread pool) y una cola de URLs pendientes), de forma que cada tarea tomará una url, la descargará, extraerá los hipervínculos encontrados en el recurso descargado, los agregará en la cola las url pendientes por descargar y vuelve a empezar. Así hasta que no haya más que hacer.
- Las tareas tienen que compartir la cola de URL pendientes y deben llevar un registro compartido de las url que se les asignan de forma que no se descargue la misma url 2 veces, ni concurrentemente (sería un caos) ni más tarde (sería un desperdicio).

Para el desarrollo de esta práctica se hizo uso de la librería threading para el uso del pool de hilos y de BeautifulSoup para el acceso a la información de la página web de donde queramos obtener la información y lo documentos.

Beautiful Soup es una biblioteca de Python para obtener datos de HTML, XML y otros lenguajes de marcado. Supongamos que se ha encontrado algunas páginas web que muestran datos relevantes para su investigación, como información de fecha o dirección, pero que no proporcionan ninguna forma de descargar los datos directamente. BeautifulSoup ayuda a extraer contenido particular de una página web, eliminar el marcado HTML y guardar la información. Es una herramienta para hacer "web scrapper" que le ayuda a limpiar y analizar los documentos que se ha extraído de la web.

Para las pruebas de este programa se hace uso de la siguiente página web: <https://www.escom.ipn.mx/>.

El funcionamiento del código es muy sencillo se ingresa la página a querer descargar el contenido y la url pasa a un proceso de verificación y obtención tanto de datos como de urls que redirigen a una página nueva, lo podemos ver como un árbol que va creciendo hacia abajo, el pool de hilo se utiliza para poder descargar todo los documentos o imágenes que se encuentren y los demás para las páginas hijas.





A continuación, mostraremos las pruebas realizadas mediante las capturas del desarrollo de la práctica.

```
Descargando HTML OrganigramaDirectorioESCOM.html [ok]
Descargando Archivo https://www.escom.ipn.mx/images/gobmxlogo.png gobmxlogo.png [OK]
Descargando Archivo https://www.escom.ipn.mx/images/logoSEP.png logoSEP.png [OK]
Descargando Archivo https://www.escom.ipn.mx/images/logoESCOM.png logoESCOM.png [OK]
Descargando Archivo https://www.escom.ipn.mx/images/logoESCOM2x.png logoESCOM2x.png [OK]
Descargando Archivo https://www.escom.ipn.mx/images/gobmxlogo_2x.png gobmxlogo_2x.png [OK]
Descargando Archivo https://www.escom.ipn.mx/images/pleca-gob.png pleca-gob.png [OK]
Link: https://www.escom.ipn.mx/htmls/escomunidad/mapaSitio.php
Link: https://www.escom.ipn.mx/htmls/conocenos/ctce.php
Link: https://www.escom.ipn.mx/htmls/conocenos/organigramaDirectorio.php
Link: https://www.escom.ipn.mx/htmls/conocenos/historiaEscudo.php
Link: https://www.escom.ipn.mx/htmls/conocenos/misionVision.php
Link: https://www.escom.ipn.mx/htmls/conocenos/organigramaDirectorio.php
Link: https://www.escom.ipn.mx/htmls/conocenos/transparencia.php
Link: https://www.escom.ipn.mx/htmls/conocenos/ubicacion.php
Link: https://www.escom.ipn.mx/htmls/oferta/isc2020.php
Link: https://www.escom.ipn.mx/htmls/oferta/isc2009.php
Link: https://www.escom.ipn.mx/htmls/oferta/iaa2020.php
Link: https://www.escom.ipn.mx/htmls/oferta/lcd2020.php
Link: https://www.escom.ipn.mx/htmls/oferta/isisa2009.php
Link: https://www.escom.ipn.mx/htmls/oferta/mcscm2018.php
Link: https://www.escom.ipn.mx/SSEIS/serviciosestudiantiles/servicios/actsCulturales.php
Link: https://www.escom.ipn.mx/SSEIS/serviciosestudiantiles/servicios/actsDeportivas.php
Link: https://www.escom.ipn.mx/SSEIS/apoyoseducativos/servicios/becas.php
Link: https://www.escom.ipn.mx/SSEIS/serviciosestudiantiles/servicios/biblioteca.php
Link: https://www.escom.ipn.mx/htmls/escomunidad/clubs.php
```

*Imagen 1 conexión*

En la siguiente imagen podemos comprobar la descarga de los archivos de la pagina de la ESCOM

docs	21/06/2023 10:04 p. m.	Carpeta de archivos
EscuelaSuperiordeCómputoIPN	21/06/2023 10:04 p. m.	Carpeta de archivos
htmls	21/06/2023 10:04 p. m.	Carpeta de archivos
images	21/06/2023 10:04 p. m.	Carpeta de archivos
MapaGeneraldelSitioWebESCOM	21/06/2023 10:04 p. m.	Carpeta de archivos
OrganigramaDirectorioESCOM	21/06/2023 10:04 p. m.	Carpeta de archivos
SSEIS	21/06/2023 10:05 p. m.	Carpeta de archivos
SubdireccióndeSEISActividadesCulturales	21/06/2023 10:04 p. m.	Carpeta de archivos
SubdireccióndeSEISActividadesDeportivas	21/06/2023 10:04 p. m.	Carpeta de archivos
SubdireccióndeSEISBecasyotrosapoyos	21/06/2023 10:05 p. m.	Carpeta de archivos
practica5	21/06/2023 09:59 p. m.	Archivo de origen ... 10 KB



## Conclusiones

Bocanegra Heziquio Yestilanezi

Fue complicado en un principio lograr descargar los archivos que estaban enlistados en la página de los servidores, porque era necesario realizar un manejo de strings de forma profunda para obtener si lo que ofrecía era un archivo o una carpeta, o en su defecto algún otro link de escape. De esta manera, de manera similar a la práctica del Drive, pudimos mapear todos los recursos que ofrecía el servidor de manera recursiva y poder ir descargando y creando las carpetas con `mkdir()` el árbol que se iba generando.

Martínez Cruz José Antonio

En esta práctica logramos comprender a mayor profundidad el uso del protocolo HTTP y como el uso del encabezado GET nos permite obtener todo el contenido que nos ofrece una página web, organizándola por carpetas y subcarpetas. Aunque existieron problemas en identificar que archivos si estaban descargados correctamente y si estaban ubicados en su lugar correspondiente, esto debido a la enorme cantidad de archivos con los que estamos lidiando. Afortunadamente logramos conseguir el objetivo de descargar los archivos que son necesarios.

